

4/5/1 (Item 1 from file: 351)
DIALOG(R) File 351:Derwent WPI
(c) 2001 Derwent Info Ltd. All rts. reserv.

012740195 **Image available**
WPI Acc No: 1999-546312/199946
XRPX Acc No: N99-405474

Graphical user interface developing apparatus - has program generating unit which produces program that displays graphical user interface component, associated with menu item, to screen of display unit and that serves as graphical user interface
Patent Assignee: SHARP KK (SHAF)
Number of Countries: 001 Number of Patents: 001
Patent Family:
Patent No Kind Date Applicat No Kind Date Week
JP 11237978 A 19990831 JP 9838684 A 19980220 199946 B

Priority Applications (No Type Date): JP 9838684 A 19980220

Patent Details:

Patent No Kind Lan Pg Main IPC Filing Notes
JP 11237978 A 13 G06F-009/06

Abstract (Basic): JP 11237978 A

NOVELTY - A program generating unit produces a program which displays a graphical user interface (GUI) component, associated with a menu item, to the screen of a display unit (1) and which serves as a graphical user interface. DETAILED DESCRIPTION - The graphical user interface developing apparatus has a menu table, a component table, the display unit, a selecting unit, and a correlating unit. The menu table consists of menu items which indicates various information processing. The component table consists of GUI components. The display unit shows each table on a screen. The selecting unit chooses a GUI component corresponding to a menu item from the tables. The correlating unit associates the chosen menu item and GUI component based on an identification code. INDEPENDENT CLAIMS are also included for the following:a GUI production; a control program for the GUI production; and a medium.

USE - For information processor e.g. personal computer, word processor.

ADVANTAGE - Easy to understand by GUI developer since correlation between a menu item and GUI component is displayed graphically and modification of correlation can be performed graphically. Improves quality of program. DESCRIPTION OF DRAWING(S) - The figure is a block diagram showing the basic components of the GUI developing apparatus.

(1) Display unit.

Dwg.1/12

Title Terms: GRAPHICAL; USER; INTERFACE; DEVELOP; APPARATUS; PROGRAM; GENERATE; UNIT; PRODUCE; PROGRAM; DISPLAY; GRAPHICAL; USER; INTERFACE; COMPONENT; ASSOCIATE; MENU; ITEM; SCREEN; DISPLAY; UNIT; SERVE; GRAPHICAL ; USER; INTERFACE

Derwent Class: P85; T01

International Patent Class (Main): G06F-009/06

International Patent Class (Additional): G06F-003/00; G09G-005/00; G09G-005/14

File Segment: EPI; EngPI

4/5/2 (Item 1 from file: 347)
DIALOG(R) File 347:JAPIO
(c) 2000 JPO & JAPIO. All rts. reserv.

06296386 **Image available**
DEVICE AND METHOD FOR PRODUCING GRAPHICAL USER INTERFACE AND MEDIUM STORING CONTROL PROGRAM

PUB. NO.: 11-237978 A]
PUBLISHED: August 31, 1999 (19990831)
INVENTOR(s): ICHIKAWA TAKEHIKO
APPLICANT(s): SHARP CORP

APPL. NO.: 10-038684 [JP'98386'84]
FILED: February 20, 1998 (19980220)
INTL CLASS: G06F-009/06; G06F-003/00; G09G-005/00; G09G-005/14

ABSTRACT

PROBLEM TO BE SOLVED: To facilitate the change and addition of association and further to improve the quality and maintainability of a program by displaying a graphical user interface part that is associated with menu items on a screen.

SOLUTION: Plural menu items that designate various information processing are stored in a menu table, graphical user interface parts are stored in a part table and each table is shown on the screen of a displaying part 1. A selecting part 2 respectively selects one menu item and a graphical user interface part corresponded to it from every table. A microcomputer 3 associates the menu item and the graphical user interface part which are selected by the part 2 based on an individually belonging identification code by using a control program stored in a medium 4, shows the associated graphical user interface part on the screen of the part 1 and produces a program executed as a graphical user interface.

COPYRIGHT: (C)1999,JPO

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-237978

(43)公開日 平成11年(1999)8月31日

(51)Int.Cl.⁶
G 0 6 F 9/06
3/00
G 0 9 G 5/00

識別記号
5 3 0
6 5 1
5 1 0

F I
G 0 6 F 9/06
3/00
G 0 9 G 5/00

審査請求 未請求 請求項の数 7 O L (全 13 頁) 最終頁に統ぐ

5 3 0 P
5 3 0 N
5 3 0 W
6 5 1 E
5 1 0 H

(21)出願番号 特願平10-38684

(71)出願人 000005049

シャープ株式会社

大阪府大阪市阿倍野区長池町22番22号

(22)出願日 平成10年(1998)2月20日

(72)発明者 市川 武彦

大阪府大阪市阿倍野区長池町22番22号 シ
ヤープ株式会社内

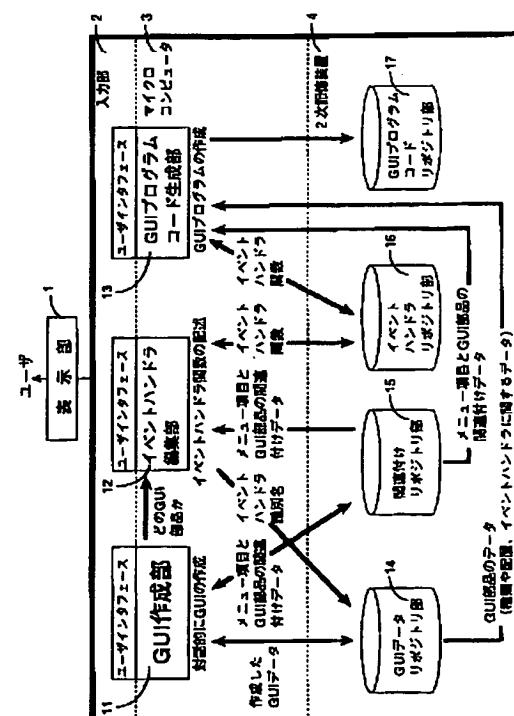
(74)代理人 弁理士 野河 信太郎

(54)【発明の名称】 グラフィカル・ユーザインタフェース作成装置及び方法並びにその制御プログラムを記憶した媒
体

(57)【要約】

【課題】 各種情報処理の指示操作において、メニュー項目と GUI 部品の関連付けが一見してわかるグラフィカル・ユーザインタフェースを作成する。

【解決手段】 各種情報処理を指示する複数のメニュー項目からなるメニューテーブルと、グラフィカル・ユーザインタフェース部品からなる部品テーブルと、前記各テーブルを画面に表示する表示部と、前記各テーブルから一つのメニュー項目とそれと対応させるグラフィカル・ユーザインタフェース部品をそれぞれ選択する選択部と、前記選択部により選択されたメニュー項目とグラフィカル・ユーザインタフェース部品とを個々に有する識別コードに基づいて関連付ける関連付け部と、前記メニュー項目に関連付いたグラフィカル・ユーザインタフェース部品を前記表示部の画面に表示しグラフィカル・ユーザインタフェースとして実行させるプログラムを生成するプログラム生成部とから構成される。



【特許請求の範囲】

【請求項1】 各種情報処理を指示する複数のメニュー項目からなるメニューテーブルと、グラフィカル・ユーザインタフェース部品からなる部品テーブルと、前記各テーブルを画面に表示する表示部と、前記各テーブルから一つのメニュー項目とそれと対応させるグラフィカル・ユーザインタフェース部品をそれぞれ選択する選択部と、前記選択部により選択されたメニュー項目とグラフィカル・ユーザインタフェース部品とを個々に有する識別コードに基づいて関連付ける関連付け部と、前記メニュー項目に関連付けたグラフィカル・ユーザインタフェース部品を前記表示部の画面に表示しグラフィカル・ユーザインタフェースとして実行させるプログラムを生成するプログラム生成部とを備えたことを特徴とするグラフィカル・ユーザインタフェース作成装置。

【請求項2】 前記プログラム生成部は、前記関連付け部により関連付けられたメニュー項目とグラフィカル・ユーザインタフェース部品間で実行するグラフィカル・ユーザインタフェースの有効あるいは無効を切り替えるプログラムを生成する機能をさらに備えたことを特徴とする請求項1記載のグラフィカル・ユーザインタフェース作成装置。

【請求項3】 前記関連付け部により関連付けられたメニュー項目またはグラフィカル・ユーザインタフェース部品のいずれか一方にショートカットキーを設定すると他方にも自動的に同じショートカットキーが設定されるショートカットキー設定部をさらに備えたことを特徴とする請求項1記載のグラフィカル・ユーザインタフェース作成装置。

【請求項4】 前記関連付け部により関連付けられたメニュー項目またはグラフィカル・ユーザインタフェース部品のいずれか一方にイベントハンドラ関数を定義すると他方にも自動的に同じイベントハンドラ関数が定義されるイベントハンドラ関数定義部をさらに備えたことを特徴とする請求項1記載のグラフィカル・ユーザインタフェース作成装置。

【請求項5】 前記関連付け部により一つのメニュー項目に複数のグラフィカル・ユーザインタフェース部品が関連付けられ、イベントハンドラ関数定義部によりグラフィカル・ユーザインタフェース部品にイベントハンドラ関数が定義される際、前記プログラム生成部は、前記メニュー項目に関連付けた複数のグラフィカル・ユーザインタフェース部品のそれぞれにイベントハンドラ関数を含むプログラムを生成することを特徴とする請求項1記載のグラフィカル・ユーザインタフェース作成装置。

【請求項6】 各種情報処理を指示する複数のメニュー項目をメニューテーブルに記憶し、グラフィカル・ユーザインタフェース部品を部品テーブルに記憶し、前記各テーブルを表示部の画面に表示し、前記各テーブルから一つのメニュー項目とそれと対応させるグラフィカル・

ユーザインタフェース部品を選択部によりそれぞれ選択し、前記選択部により選択されたメニュー項目とグラフィカル・ユーザインタフェース部品とを個々に有する識別コードに基づいて関連付けし、前記メニュー項目に関連付けたグラフィカル・ユーザインタフェース部品を前記表示部の画面に表示しグラフィカル・ユーザインタフェースとして実行させるプログラムを生成するグラフィカル・ユーザインタフェース作成方法。

【請求項7】 各種情報処理を指示する複数のメニュー項目からなるメニューテーブルと、グラフィカル・ユーザインタフェース部品からなる部品テーブルと、前記各テーブルを画面に表示する表示部と、前記各テーブルから一つのメニュー項目とそれと対応させるグラフィカル・ユーザインタフェース部品をそれぞれ選択する選択部とを備えたグラフィカル・ユーザインタフェース装置をコンピュータによって制御するプログラムを記憶した媒体であって、前記コンピュータに、前記選択部により選択されたメニュー項目とグラフィカル・ユーザインタフェース部品とを個々に有する識別コードに基づいて関連付けさせ、前記メニュー項目に関連付けたグラフィカル・ユーザインタフェース部品を前記表示部の画面に表示しグラフィカル・ユーザインタフェースとして実行させるプログラムを生成させるグラフィカル・ユーザインタフェース作成装置制御プログラムを記憶した媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、パーソナルコンピュータ、ワードプロセッサなどの情報処理機器において、ユーザが視覚的、直感的に情報処理を実行するためのグラフィカル・ユーザインタフェースを作成するグラフィカル・ユーザインタフェース作成装置及び方法並びにその制御プログラムを記憶した媒体に関する。

【0002】

【従来の技術】 近年、情報処理機器のインターフェースとして、ユーザに視覚的、直感的にわかりやすく、かつ使いやすいグラフィカル・ユーザインタフェース（以下、G U I という）が採用されることが多くなっている。

【0003】 表示画面のウィンドウの一部分に、常時あるいは必要に応じてメニューインドウが表示され、その各メニュー項目に対して情報処理機器上で動作するソフトウェアにより機能が割り当てられている。従って、ユーザは、通常、これらのメニュー項目をマウスで表示画面上を操作して選択することにより所望の G U I の機能を実行させている。

【0004】 また、メニュー項目に割り当てられている機能の選択および実行が手軽に行えるように、メニュー項目の選択の煩雑さを解消している。例えば、使われる頻度が高い機能が割り当てられたメニュー項目については、ウィンドウに表示されるキー、ボタンなどのグラフィカル・ユーザインタフェース部品（以下、G U I 部品

という)に関連付けられることがよく行われる。G U I 部品とは、ユーザが目的の情報処理の項目を指示して実行するためのアイコン、キー、ボタンなどの部品をいう。

【0005】さらに、情報処理機器の操作に熟練したユーザのために、メニュー項目の一部にショートカットキーを設定し、メニュー項目に割り当てた機能を、キーボードで選択し、実行できるように構成されている。ショートカットキーとは、通常、一つのキーに別の目的の機能を持たせたことをいう。

【0006】ある一つのメニュー項目と関連付けられたG U I 部品、例えば、それがボタンであれば、マウスでそのボタンをクリックしたときは、関連付けられているメニュー項目を選択したときと同じ機能が実行される。また、メニュー項目の選択操作の有効・無効の切り替えと同期して、関連付けられているG U I 部品の操作の有効・無効も切り替えられる。さらに、ショートカットキーがメニュー項目に設定されている場合には、関連付けられているG U I 部品にも同じショートカットキーが設定されることが通常である。

【0007】従って、従来のG U I 作成装置では、メニュー項目とG U I 部品の関連付けを、次のようにして実現している。例えば、メニュー項目及びG U I 部品のいずれか一方のイベントハンドラ関数を呼び出すように、他方のイベントハンドラ関数を記述する。また、メニュー項目及びG U I 部品のいずれか一方の操作の有効・無効を切り替えた場合は、他方の操作の有効・無効もあわせて切り替えるようにプログラムコードを生成する。また、ショートカットキーは、両方それぞれに設定する。イベントハンドラとは、マウスによるメニュー項目の選択やボタンのクリックなどのG U I へのユーザの一つの操作の区切り(イベント)に対してメニュー項目やG U I 部品ごとに実行される処理をいう。

【0008】

【発明が解決しようとする課題】しかしながら、従来のG U I 作成装置でG U I を作成した際、次のような問題がある。

(1) どのメニュー項目とG U I 部品が関連付けられているのか、ユーザには一見してわからない。これは、メニュー項目とG U I 部品との関連付けにおいて、

- ・同じイベントハンドラ関数を呼び出す構成になっている。

- ・操作の有効・無効を同時に切り替えるプログラムを記述する構成になっている。

- ・同じショートカットキーを設定する構成になっている。

など、プログラム中のさまざまな部分で行われているためである。そのため、プログラムをある程度解析しなければならない。このため、関連付けを行うためのプログラムコードにバグが発生する確率を高める要因となり、

また、後からのプログラムの変更、追加等の保守が困難になる。

【0009】(2)さらに、一つのメニュー項目に関連付けられた全てのG U I 部品について、単に関連付けるためだけのイベントハンドラ関数を定義することになる。また、操作の有効・無効の切り替えのためのプログラムコードの作成やショートカットキーの設定も、関連付けられている全てのG U I 部品について必要になる。

これは、プログラミングする際のプログラムコードやデータの量を増加させることになり、これらの関連付けを行うためのプログラムコードにバグが発生する確率も高める要因となる。

【0010】本発明は以上の事情を考慮してなされたものであり、例えば、各種情報処理の指示操作において、ユーザがメニュー項目とG U I 部品の関連付けが一見してわかるグラフィカル・ユーザインタフェースを作成して利用することができるグラフィカル・ユーザインタフェース作成装置及び方法並びにその制御プログラムを記憶した媒体を提供する。

【0011】

【課題を解決するための手段】本発明は、各種情報処理を指示する複数のメニュー項目からなるメニューテーブルと、グラフィカル・ユーザインタフェース部品からなる部品テーブルと、前記各テーブルを画面に表示する表示部と、前記各テーブルから一つのメニュー項目とそれと対応させるグラフィカル・ユーザインタフェース部品をそれぞれ選択する選択部と、前記選択部により選択されたメニュー項目とグラフィカル・ユーザインタフェース部品とを個々に有する識別コードに基づいて関連付けする関連付け部と、前記メニュー項目に関連付けたグラフィカル・ユーザインタフェース部品を前記画面に表示しグラフィカル・ユーザインタフェースとして実行させるプログラムを生成するプログラム生成部とを備えたことを特徴とするグラフィカル・ユーザインタフェース作成装置である。

【0012】なお、本発明において、メニューテーブルは、各種情報処理内容を示す複数のメニュー項目からなるメニューバー、メニューウィンドウなどを示し、メニューテーブルの各データはR O M、フロッピーディスクなどの記憶部に記憶されている。部品テーブルは、ユーザが目的の情報処理を図形的に指示して実行するためのグラフィカル・ユーザインタフェース部品として、アイコン、キー、ボタンなど部品からなるパレットやウィンドウを示し、部品テーブルの各データはR O M、フロッピーディスクなどの記憶部に記憶されている。

【0013】表示部は、LCD(液晶ディスプレイ)、P D (プラズマ・ディスプレイ)などで構成することができる。選択部は、マウス、ペン、トラックパッドなどのポインティングデバイスで構成することができる。関連付け部、プログラム生成部は、C P U、R O M、R A

M、I/Oポートからなるマイクロコンピュータで構成することができる。

【0014】本発明によれば、各種情報処理の指示操作において、ユーザはメニュー項目とGUI部品の関連付けが一見してわかるグラフィカル・ユーザインタフェースを作成して利用することができる。

【0015】前記プログラム生成部は、前記関連付け部により関連付けられたメニュー項目とグラフィカル・ユーザインタフェース部品間で実行するグラフィカル・ユーザインタフェースの有効あるいは無効を切り替えるプログラムを生成する機能をさらに備えた構成にしてもよい。前記構成によれば、関連付けられているメニュー項目とGUI部品間での有効あるいは無効の切り替えが自動的に行われる所以、プログラムの変更、追加が簡便になる。

【0016】前記関連付け部により関連付けられたメニュー項目またはグラフィカル・ユーザインタフェース部品のいずれか一方にショートカットキーを設定すると他方にも自動的に同じショートカットキーが設定されるショートカットキー設定部をさらに備えた構成にしてもよい。なお、ショートカットキー設定部は、マイクロコンピュータで構成することができる。前記構成によれば、関連付けられているGUI部品あるいはメニュー項目があれば、そのショートカットキーを設定するべきプロパティに自動的に同じ設定が行われるので、ショートカットキーを設定するプログラムの変更、追加が簡便になる。

【0017】前記関連付け部により関連付けられたメニュー項目またはグラフィカル・ユーザインタフェース部品のいずれか一方にイベントハンドラ関数を定義すると他方にも自動的に同じイベントハンドラ関数が定義されるイベントハンドラ関数定義部をさらに備えた構成にしてもよい。なお、イベントハンドラ関数定義部は、マイクロコンピュータで構成することができる。前記構成によれば、関連付けられているメニュー項目やGUI部品間でイベントハンドラが共有できる。

【0018】前記関連付け部により一つのメニュー項目に複数のグラフィカル・ユーザインタフェース部品が関連付けられ、イベントハンドラ関数定義部によりグラフィカル・ユーザインタフェース部品にイベントハンドラ関数が定義される際、前記プログラム生成部は、前記メニュー項目に関連付けた複数のグラフィカル・ユーザインタフェース部品のそれぞれにイベントハンドラ関数を含むプログラムを生成する構成にしてもよい。前記構成によれば、関連付けられているメニュー項目やGUI部品ごとのイベントハンドラ関数を定義することができ、イベントハンドラ関数を定義するプログラムの変更、追加が簡便になる。

【0019】

【発明の実施の形態】以下、図に示す実施例に基づいて

本発明を詳述する。なお、本発明はこれによって限定されるものではない。

【0020】図1は本発明のグラフィカル・ユーザインタフェース作成装置の基本構成を示すブロック図である。グラフィカル・ユーザインタフェースは以下、GUIという。図1において、1はLCD(液晶ディスプレイ)、PD(プラズマ・ディスプレイ)などで構成される表示部を示す。2はマウス、ペン、トラックパッドなどのポインティングデバイスで構成される入力部を示す。10 入力部2は、選択部、設定部などの各入力部としても機能する。

【0021】3はCPU、ROM、RAM、I/Oポートからなるマイクロコンピュータを示し、マイクロコンピュータ3は、GUI作成部(関連付け部)、GUIプログラムコード生成部(プログラム生成部)、イベントハンドラ編集部(イベントハンドラ関数定義部)、及びショートカットキー設定部として機能する。4はEEPROMからなるICカード、フロッピーディスク、ハードディスク、CD-ROM、MDなどの記憶媒体で構成される2次記憶装置を示す。2次記憶装置4には、マイクロコンピュータ3が、GUI作成装置の各部として機能する制御プログラムが記憶するようにしてよい。

【0022】14はGUI部品データ(作成されたGUIデータの種類や配置、イベントハンドラに関するデータ)を保存するGUIデータリポジトリ部を示し、2次記憶装置4の領域に含まれる。また、GUIデータリポジトリ部14は、表示部1の画面に表示するGUI部品の部品テーブル、複数のメニュー項目のメニューテーブルとして機能する。15はメニュー項目とGUI部品の関連付けデータを保存する関連付けリポジトリ部を示し、2次記憶装置4の領域に含まれる。

【0023】11は、ユーザ(GUI開発者)が、GUIデータリポジトリ部14に保存したメニュー項目やGUI部品のデータと関連付けリポジトリ部15に保存した関連付けデータに基づいて対話的にGUIデータを作成するためのGUI作成部を示す。マイクロコンピュータ3がGUI作成プログラムによりGUI作成部11として機能する。16はイベントハンドラ関数を格納するイベントハンドラリポジトリ部を示し、2次記憶装置4の領域に含まれる。

【0024】12はGUI作成部11に目的のGUI部品を問い合わせ、関連付けリポジトリ部5に保存したメニュー項目及びGUI部品の関連付けデータとイベントハンドラリポジトリ部16に保存したイベントハンドラ関数に基づいてイベントハンドラに関するデータ(例えば、イベントハンドラ識別名)の編集を行うイベントハンドラ編集部を示す。マイクロコンピュータ3がイベントハンドラ編集プログラムによりイベントハンドラ編集部12として機能する。また、GUI作成部11は、前に作成したGUIデータ、イベントハンドラ編集部12

で編集されたイベントハンドラ識別名をG U I データリポジトリ部14から読み込むことができる。

【0025】13はG U I データリポジトリ部14に保存されたG U I 部品データ、関連付けリポジトリ部15に保存された関連付けデータ、イベントハンドラリポジトリ部16に保存されたイベントハンドラ関数に基づいてG U I プログラムコードを生成するG U I プログラムコード生成部を示す。マイクロコンピュータ3がG U I プログラムコード生成プログラムによりG U I プログラムコード生成部13として機能する。17はG U I プログラムコードを格納するG U I プログラムコードリポジトリ部を示す。

【0026】図1に示す構成において、本発明は、各種情報処理を指示する複数のメニュー項目をメニュー一テーブルに記憶し、グラフィカル・ユーザインタフェース部品を部品テーブルに記憶し、前記各テーブルを表示部1の画面に表示し、前記各テーブルから一つのメニュー項目とそれと対応させるグラフィカル・ユーザインタフェース部品を選択部2によりそれぞれ選択し、前記選択部2により選択されたメニュー項目とグラフィカル・ユーザインタフェース部品とを個々に有する識別コードに基づいて関連付けし、前記メニュー項目に関連付けたグラフィカル・ユーザインタフェース部品を前記画面に表示しグラフィカル・ユーザインタフェースとして実行させるプログラムを生成するグラフィカル・ユーザインタフェース作成方法を提供することができる。

【0027】さらに、本発明は、各種情報処理を指示する複数のメニュー項目からなるメニュー一テーブルと、グラフィカル・ユーザインタフェース部品からなる部品テーブルと、前記各テーブルを画面に表示する表示部1と、前記各テーブルから一つのメニュー項目とそれと対応させるグラフィカル・ユーザインタフェース部品をそれぞれ選択する選択部2とを備えたグラフィカル・ユーザインタフェース装置をマイクロコンピュータ3によって制御するプログラムを記憶した媒体4であって、前記マイクロコンピュータ3に、前記選択部2により選択されたメニュー項目とグラフィカル・ユーザインタフェース部品とを個々に有する識別コードに基づいて関連付けさせ、前記メニュー項目に関連付けたグラフィカル・ユーザインタフェース部品を前記表示部1の画面に表示しグラフィカル・ユーザインタフェースとして実行させるプログラムを生成させるグラフィカル・ユーザインタフェース装置制御プログラムを記憶した媒体4を提供することができる。

【0028】図2は本実施例におけるG U I 作成処理前の画面例を示す説明図である。図2に示すように、G U I 開発者(ユーザ)は、G U I 作成部11を起動させて、G U I 作成画面を表示する。例えば、マウスでG U I 部品パレット21から関連付けたいG U I 部品(例えば、ボタン“OK”)を選び、G U I メニュー22のメ

ニュー項目(例えば、“開く”)上にドラッグ・アンド・ドロップ操作をすることで、G U I 作成部11は、メニュー項目(例えば、“開く”)とG U I 部品(例えば、ボタン“OK”)をグラフィカルに関連付ける。その後、関連付けたG U I 部品を画面上の適切な場所に配置する。関連付けるG U I 部品としては、G U I 部品パレット21から選択してもよいし、新規に作成してもよい。また、すでに作成され、G U I データリポジトリ部14に保持されているG U I 部品を取り出してもよい。

10 【0029】図3は本実施例におけるG U I 作成処理後の画面例を示す説明図である。図3において、31は個々のメニュー項目やG U I 部品を編集するための編集メニューを示す。G U I 開発者は、メニュー項目(“開く”)とG U I 部品(ボタン“OK”)とを関連させた後、G U I 部品やメニュー項目を選択状態にして表示する。例えば、マウスの右ボタンを押下すれば、編集メニュー31が開くように構成することにより、メニュー項目とG U I 部品の関連付けを確認する。

【0030】そして、編集メニュー31を開いて、例え20ば“関連表示”を選択すると、関連付けられているメニュー項目あるいはG U I 部品との間に直線が表示されるようになる。さらに、編集メニュー31の“関連削除”を選択することによって、関連付けを解除できるように構成してもよい。

【0031】メニュー項目やG U I 部品間の関連付けの情報は、メニュー項目やG U I 部品パレット22のG U I 部品ごとにG U I 作成部11で一意に付される識別名とともに、関連付けリポジトリ部15に保存される。一つの関連付けを表す関連付けデータの形式としては、例30えば、次のような組とすればよい。

【0032】(MENUITEMO, BUTTONO,.....) “MENUITEMO”は、メニュー項目の識別名を示す。“BUTTONO”は、メニュー項目(MENUITEMO)に関連付けたG U I 部品の識別名を示す。このように関連付けられた全ての識別名を、一括して関連付けリポジトリ15に格納する。

【0033】G U I データリポジトリ部14には、G U I 開発者が、G U I 作成部11で作成したG U I 情報、すなわち、メニュー項目やG U I 部品の個々のデータ(位置やサイズ、キャプション、G U I 開発者が付けた名前、ショートカットキーの設定などのプロパティの値)が保持される構成とする。

【0034】そして、各データにはG U I 作成部11で作成した識別名を付して、識別名をキーとして検索できる構成にする。このように構成すれば、G U I 作成部11は、G U I データリポジトリ部14と関連付けリポジトリ部15からデータを読み込んで照合することにより、一旦作成し保存したG U I データ及び関連付けデータを再現することができる。

50 【0035】G U I 開発者が、メニュー項目やG U I 部

品に割り当てたショートカットキーは、メニュー項目やG U I 部品のプロパティとして個々に設定されることが一般的である。よって、G U I 開発者によってショートカットキーの設定が行われたら、G U I 作成部1 1は、関連付けリポジトリ部1 5を参照して、関連付けられているG U I 部品あるいはメニュー項目がないかを検索する。そして、もし関連付けられているG U I 部品あるいはメニュー項目があれば、そのショートカットキーを設定するべきプロパティに同じ設定を行うように構成する（図1 1参照）。

【0036】イベントハンドラ関数を定義するときには、まず、“どのメニュー項目あるいはG U I 部品の、どのイベントに対するイベントハンドラ関数か”を指定する必要がある。イベントには、マウスのクリックやマウスカーソルの進入、G U I のオープンやクローズ、情報処理機器からのタイマや電源オフによる割り込みなどがある。これらのイベントが発生したときにG U I が行うべき動作を記述したものがイベントハンドラ関数である。このイベントハンドラ関数も、プロパティの値と同様に、個々のメニュー項目やG U I 部品の一つのデータとして登録される。

【0037】図4は本実施例におけるG U I のイベントハンドラ編集処理（1）の画面例を示す説明図である。図4において、4 1はイベントハンドラ関数を定義するイベントを選択するためのイベントリストを示す。本実施例のG U I 作成装置では、G U I 作成部1 1で、G U I 開発者が、図3において、編集メニュー3 1の“イベントハンドラ編集”を選択すると、そのとき選択状態にあるメニュー項目あるいはG U I 部品のイベントリスト4 1が開く。G U I 開発者は、このイベントリスト4 1から、イベントハンドラ関数を定義あるいは編集したいイベントを選択する。

【0038】G U I 開発者は、イベントを選択したら、イベントハンドラ編集部1 2を起動する。イベントハンドラ編集部1 2でイベントハンドラ関数を編集を行う。さらに、関連付けリポジトリ部1 5からG U I 作成部1 1で関連付けられた他のメニュー項目あるいはG U I 部品を検索して、現在、編集しているイベントハンドラ関数を、どのイベントハンドラ関数にするか共有するなどを選択できるように構成する。ただし、デフォルトとして、もっとも一般的なイベントハンドラ関数が自動的に登録される構成にする。

【0039】図5は本実施例のG U I のイベントハンドラ編集処理（2）の画面例を示す説明図である。図5において、5 1はイベントハンドラ関数を編集するためのテキストエディタを示す。5 2は編集中のイベントハ

```
MENUOITEM0 As MenuItem
{
    Property {
        .....
    }
}
```

ドラ関数を共有したいG U I 部品とそのイベントを選択するためのメニューを示す。G U I 開発者は、このテキストエディタ5 1の【関連選択】メニューでG U I 作成部1 1で関連付けた他のメニュー項目あるいはG U I 部品を選択する（この例ではメニュー項目“MENU0ITEM0”）。そして、さらに編集中のイベントハンドラ関数を共有するイベントを選択する。この例では、デフォルトでイベント“SELECTED”が選択されている。

【0040】イベントハンドラ編集部1 3は、G U I 開発者が定義したイベントハンドラ関数に一意な識別名を割り当てる。そして、識別名とイベントハンドラ関数の組をイベントハンドラリポジトリ部1 6に格納する。例えば、次のような形式で保存する。

【0041】

```
EVENTHANDLER000 =20
int FileOpen(String fname)
{
    .....
}
```

【0042】ここで、“EVENTHANDLER000”は、イベントハンドラ編集部が自動的に付した、イベントハンドラ関数（この例では、“FileOpen()”関数）を一意に識別するためのイベントハンドラ識別子を示す。“=20”は、“int FileOpen(String fname)”の行から“}”までの20行がイベントハンドラ関数のソースプログラムであることを意味する。このソースプログラムの部分は、最終的にG U I プログラムコード生成部1 3によって抽出され、G U I プログラムコードとして記述される。イベントハンドラリポジトリ部1 6では、イベントハンドラ識別名とソースプログラムのセットで一つのイベントハンドラ関数の情報が保存される。

【0043】さらに、現在編集中のイベントハンドラ関数を共有するもの全て（すなわち、関連付けられているメニュー項目やG U I 部品も含めて）に、G U I 開発者によってあるいはデフォルトで選択されるイベントのイベントハンドラ関数として、イベントハンドラ編集部が割り当てた識別名をG U I データリポジトリ1 4に格納する。例えば、関連付けられているメニュー項目“MENU0ITEM0”とボタン“BUTTON0”があり、“MENU0ITEM0”的イベント“SELECTED”と“BUTTON0”的イベント“CLICK”に、同じイベントハンドラ関数“EVENTHANDLER000”が設定されたとき、G U I データリポジトリ1 4には、例えば次のような形式で格納される。

【0044】

… (1)

… (2)

```

        name="ファイルを開く(&O)"
        .....
        short-cut-key=Ctrl+O
        .....
    }
Event {
    .....
    CLICK=EVENTHANDLER000
    .....
}
}
.....
... (3)
... (4)
... (5)
... (6)

BUTTON0 As CommandButton
{
Property {
    .....
    name= "OK"
    short-cut-key=Ctrl+O
    .....
}
Event {
    .....
    CLICK=EVENTHANDLER000
    .....
}
}

```

【0045】(1)～(6)の部分が、識別名MENUITEM0のメニュー項目あるいはボタンなどのG U I 部分のデータを示す。(1)の“As MenuItem”の部分で、このメニュー項目あるいはボタンなどのG U I 部分の種類を宣言している(例えば、メニュー項目だとかボタン、リスト、テキスト表示領域など)。(2)～(3)の部分は、MENUITEM0のプロパティ名とその値を示す。本実施例ではショートカットキーはプロパティに設定しており、プロパティshort-cut-keyの値“Ctrl+O”として表される。(4)～(5)の部分は、イベントとそれに対応する(設定されているイベントハンドラ関数の)イベントハンドラ識別名を示す。上記の例では、イベントCLICKにイベントハンドラ識別名EVENTHANDLER000が設定されている。G U I プログラムコード生成部分13は、ここで指定されているイベントハンドラ識別名を基に、イベントハンドラリポジトリ部16から同じイベントハンドラ識別名を持つイベントハンドラ関数のソースプログラムを抽出する。

【0046】図6は実施例1におけるG U I 関連付け作成処理の手順を示すフローチャートである。図6において、

ステップS61：G U I 関連付け作成処理を開始する。
ステップS62：ドラッグされているG U I 部品の識別名GUIDをG U I 作成部11により取得する。

ステップS63：ドロップされたメニュー項目の識別名MENUIDをG U I 作成部11により取得する。

30 ステップS64：関連付けリポジトリ部15にMENUIDとGUIDの関連付けを登録する。

ステップS65：G U I 関連付け作成処理終了。

【0047】図7は実施例1におけるG U I 関連付け表示処理の手順を示すフローチャートである。図7において、

ステップS71：G U I 関連付け表示処理を開始する。

ステップS72：G U I 作成部11において、選択されているG U I 部品あるいはメニュー項目別名“SELECTED ID”を取得する。

40 ステップS73：関連付けリポジトリ部15から識別名“SELECTEDID”に関連付けられているG U I 部品やメニュー項目の識別名をすべて得る。

ステップS74：G U I 作成部11において、選択されているG U I 部品あるいはメニュー項目から、関連付けリポジトリ部15から取得された全ての識別名に対応するG U I 部品あるいはメニュー項目に対して接続する直線を描画する。

ステップS75：G U I 関連付け表示処理終了。

【0048】図8は実施例1におけるG U I 関連付け削除処理の手順を示すフローチャートである。図8において、

ステップS 8 1 : G U I 関連付け削除処理を開始する。
 ステップS 8 2 : G U I 作成部1 1において、関連付けられたG U I 部品あるいはメニュー項目を選択して識別名GUID、MENUIDを得る。
 ステップS 8 3 : 関連付けリポジトリ部1 5から、GUIDとMENUIDの関連付けを削除する。
 ステップS 8 4 : G U I 作成部1 1で選択されている関連付け(直線)を削除する。

ステップS 8 5 : G U I 関連付け削除処理終了。

【0049】前記実施例1の構成によれば、G U I メニューの各メニュー項目とG U I 部品をグラフィカルに関連付けることが実現できるので、ユーザは、一見してメニュー項目とG U I 部品の関連付けがわかる。

【0050】G U I プログラムコード生成部1 3では、G U I データリポジトリ部1 4、関連付けリポジトリ部1 5、およびイベントハンドラリポジトリ部1 6に保存されているデータを基に、ターゲット原語(Java, C++, Cなど)で書かれたプログラムコードを生成する。

【0051】このとき、G U I プログラムコード生成部1 3は関連付けリポジトリ1 5の関連付けデータを基に、関連付けられているメニュー項目やG U I 部品の有効・無効をいっせいに切り替えるための関数“_validateRelationalGroup()”と、識別名を与えると直接・間接に関連付けられているメニュー項目やG U I 部品の識別名のリストに返す“_getRelationalGroup()”を作成する。

【0052】メニュー項目やG U I 部品の有効・無効の切り替えは、それぞれの特定のプロパティの値を変更したり、あるいはそれぞれに関数を呼出してやることが一般的である。本実施例では、あるG U I 部品“BUTTON0”およびそれに直接・間接に関連付けられているすべてのメニュー項目やG U I 部品すべての有効・無効を切り替えたいときは、G U I 開発者は、“_validateRelationalGroup(BUTTON0)”を呼び出すプログラムを記述する。

【0053】G U I プログラムコード生成部1 3は、“_validateRelationalGroup()”内で、“_getRelationalGroup(BUTTON0)”を呼び出し、識別名のリストを得るようにプログラムコードを生成するようにしておく、このために関連付けリポジトリのデータを基に、関数“_getRelationalGroup()”内に関連付けられているメニュー項目やG U I 部品の識別名のテーブルを作成するプログラムコードも生成するように構成する。そして、関数“_validateRelationalGroup()”内では、得られた識別名のリストの各要素に対して、有効・無効の切り替えを行うように構成する。

【0054】図9は実施例2におけるG U I の有効・無効切り替えプログラム生成処理(1)の手順を示すフローチャートである。図9において、
 ステップS 9 1 : G U I の有効・無効切り替えプログラム生成処理開始。

ステップS 9 2 : 関連付けられているメニュー項目やG U I 部品の有効・無効を切り替える関数“_validateRelationalGroup()”のプログラムコードを書き出す。

ステップS 9 3 : G U I データリポジトリ部1 4から、全ての識別名を取得する。

ステップS 9 4 : 関連付けリポジトリ部1 5から全ての関連付け情報を取得する。

ステップS 9 5 : ある識別名を与えられたら、関連付けられている全てのメニュー項目あるいはG U I 部品の識別名のリストに返す関数“_getRelationalGroup()”のプログラムコードを記述する。

ステップS 9 6 : G U I の有効・無効切り替えプログラム生成処理終了。

【0055】図10は実施例2におけるG U I の有効・無効切り替えプログラム生成処理(2)の手順を示すフローチャートである。図10において、

ステップS 1 0 1 : “_validateRelationalGroup()”のプログラムコードの生成処理開始。

ステップS 1 0 2 : 引数に与えられた識別名を引数に“_getRelationalGroup()”を呼び出し、関連付けられている識別名のリストを取得する。

ステップS 1 0 3 : 取得したリスト中の全ての識別名を処理したか判定する。

ステップS 1 0 4 : ステップS 1 0 3で、全ての識別名を処理していないければ、リストから取り出した識別名のメニュー項目あるいはG U I 部品の有効・無効を切り替える。

ステップS 1 0 5 : ステップS 1 0 3で、全ての識別名を処理しておれば、“_validateRelationalGroup()”のプログラムコードの生成処理終了。

【0056】前記実施例2の構成によれば、関連付けられているメニュー項目とG U I 部品間での有効あるいは無効の切り替えが自動的に行われる所以、プログラムの変更、追加が簡便になる。

【0057】図11は実施例3におけるG U I のショートカットキー自動設定処理の手順を示すフローチャートである。図11において、

ステップS 1 1 1 : G U I のショートカットキー自動設定処理開始。

ステップS 1 1 2 : G U I 作成部1 1により、ショートカットキーを設定されたメニュー項目あるいはG U I 部品の識別名と、そのショートカットキーの値を取得する。関連付けられているメニュー項目やG U I 部品の有効・無効を切り替える関数“_validateRelationalGroup()”のプログラムコードを記述する。

【0058】ステップS 1 1 3 : 関連付けリポジトリ部1 5から、先に得られた識別名と関連付けられている識別名のリストを取得する。G U I データリポジトリ部1 4から、全ての識別名を取得する。

ステップS 1 1 4 : リスト中の全ての識別名について処理

が終了したか判定する。

ステップS115：ステップS114で、全ての識別名について処理が終了していないければ、リストから取り出した識別名のメニュー項目あるいはGUI部品に、同じショートカットキーを設定して、GUIデータリポジトリ部14に保存する。

ステップS116：ステップS114で、全ての識別名について処理が終了しておれば、GUIのショートカットキー自動設定処理終了。

【0059】前記実施例3の構成によれば、関連付けられたメニュー項目とGUI部品でショートカットキーの設定が共有できる。

【0060】図12は実施例4におけるGUIのイベントハンドラ関数自動設定処理の手順を示すフローチャートである。図12において、

ステップS121：GUIのイベントハンドラ関数自動設定処理開始。

ステップS122：イベントハンドラ編集部12で、イベントハンドラ関数を設定されたメニュー項目あるいはGUI部品の識別名とイベント名、及びそのイベントハンドラ関数の識別名（＝イベントハンドラ識別名）を取得する。

ステップS123：関連付けリポジトリ部15から、先に得られた識別名と関連付けられている識別名のリストを得る。

【0061】ステップS124：リスト中の全ての識別名について処理が終了したか判定する。

ステップS125：ステップS124で、全ての識別名について処理が終了していないければ、リストから取り出した識別名のメニュー項目あるいはGUI部品に該当するイベントに、同じイベントハンドラ識別名を設定して、GUIデータリポジトリ部14に保存する。

ステップS126：ステップS114で、全ての識別名について処理が終了しておれば、GUIのイベントハンドラ関数自動設定処理終了。

【0062】前記実施例4の構成によれば、関連付けられているメニュー項目やGUI部品間でイベントハンドラの共有が実現できる。

【0063】関連付けられているメニュー項目やGUI部品間で、イベントハンドラ関数のはほとんど部分を共有できるときでも、個々のGUI部品独自の処理（例えば、特定のプロパティの設定など）が少しであると共有できない。そこで、イベントハンドラ関数内で関連付けられているメニュー項目やGUI部品を個々に識別し、固有の処理を記述できるようにしておく。GUIプログラムコード生成部3が作成する関数“_getRelationalGroup()”によって、GUI開発者は、この識別名を用いて分岐処理を記述することで、個々のメニュー項目やGUI部品ごとの処理を記述することができる。

【0064】イベントハンドラ編集部12で、直接・間

接に関連しているメニュー項目やGUI部品間でイベントハンドラ関数の共有が行える。また、共有しているイベントハンドラ関数内で個々のGUI部品ごとの処理も記述できるので、共有できるイベントハンドラ関数は増える。さらに、個々の直接・間接に関連しているメニュー項目やGUI部品の有効・無効を一度に切り替えられる手段も提供される。これらのこととは、従来に比べ、GUIを実現するプログラムの量を減らすことができる。プログラムの量が減れば、その分バグは減るので、プログラムの信頼性が向上する。

【0065】

【発明の効果】本発明によれば、メニュー項目やGUI部品間での関連付けがグラフィカルに表示されるので、GUI開発者にとって把握しやすい。また、関連付けの変更、追加もグラフィカルに行えるので容易である。さらに、プログラムの品質、保守性が向上する。

【図面の簡単な説明】

【図1】本発明のグラフィカル・ユーザインタフェース作成装置の基本構成を示すブロック図である。

20 【図2】本実施例におけるGUI作成処理前の画面例を示す説明図である。

【図3】本実施例におけるGUI作成処理後の画面例を示す説明図である。

【図4】本実施例におけるGUIのイベントハンドラ編集処理（1）の画面例を示す説明図である。

【図5】本実施例におけるGUIのイベントハンドラ編集処理（2）の画面例を示す説明図である。

【図6】実施例1におけるGUI関連付け作成処理の手順を示すフローチャートである。

30 【図7】実施例1におけるGUI関連付け表示処理の手順を示すフローチャートである。

【図8】実施例1におけるGUI関連付け削除処理の手順を示すフローチャートである。

【図9】実施例2におけるGUIの有効・無効切り替えプログラム生成処理（1）の手順を示すフローチャートである。

【図10】実施例2におけるGUIの有効・無効切り替えプログラム生成処理（2）の手順を示すフローチャートである。

40 【図11】実施例3におけるGUIのショートカットキー自動設定処理の手順を示すフローチャートである。

【図12】実施例4におけるGUIのイベントハンドラ関数自動設定処理の手順を示すフローチャートである。

【符号の説明】

1 表示部

2 入力部

3 マイクロコンピュータ

4 2次記憶装置（記憶媒体）

11 GUI作成部

50 12 イベントハンドラ編集部

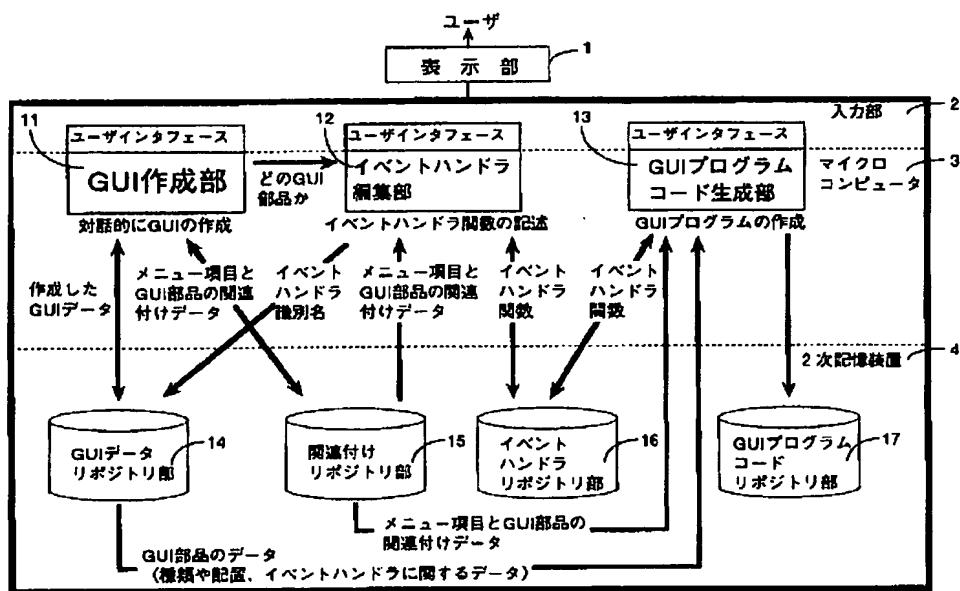
17

- 1 3 GUIプログラムコード生成部
 1 4 GUIデータリポジトリ部
 1 5 関連付けリポジトリ部

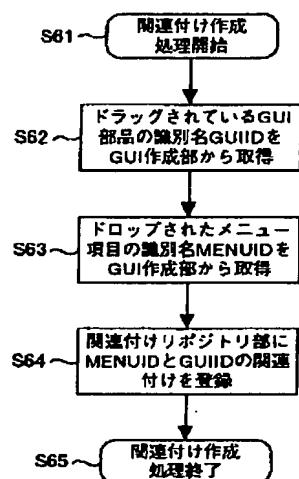
18

- 1 6 イベントハンドラリポジトリ部
 1 7 GUIプログラムコードリポジトリ部

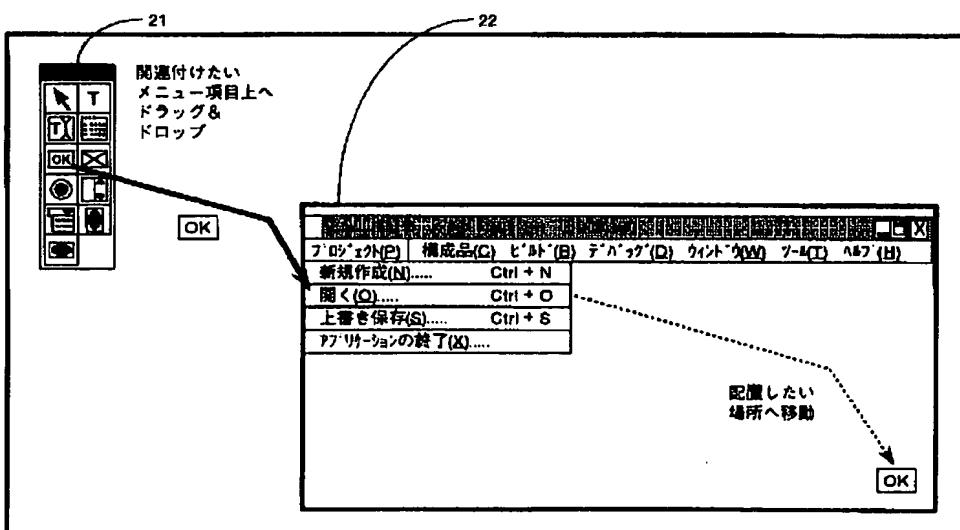
【図1】



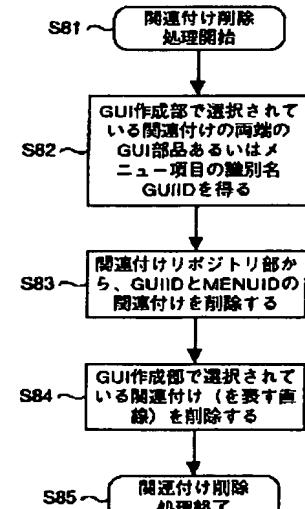
【図6】



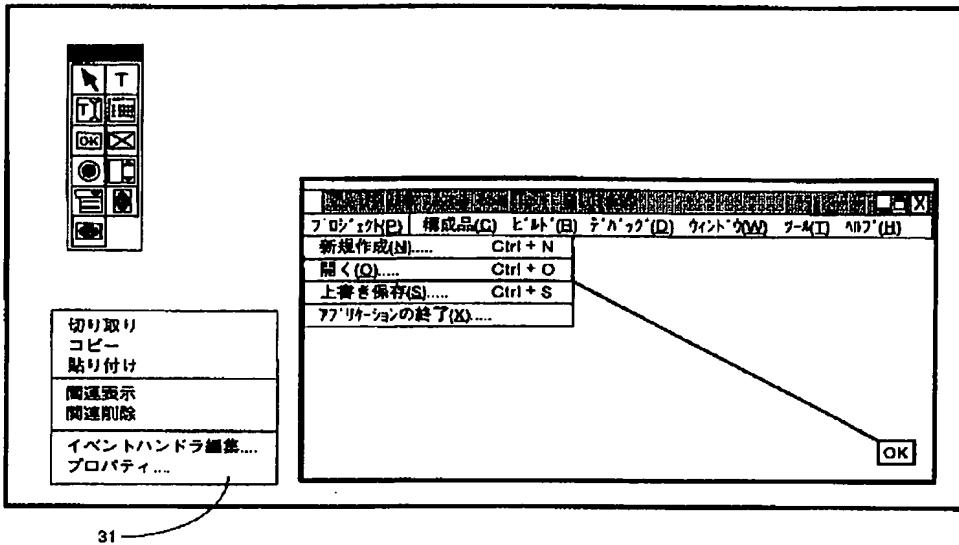
【図2】



【図8】

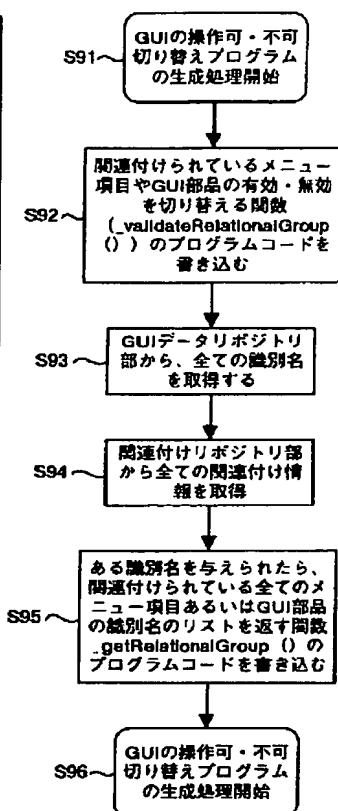


【図3】

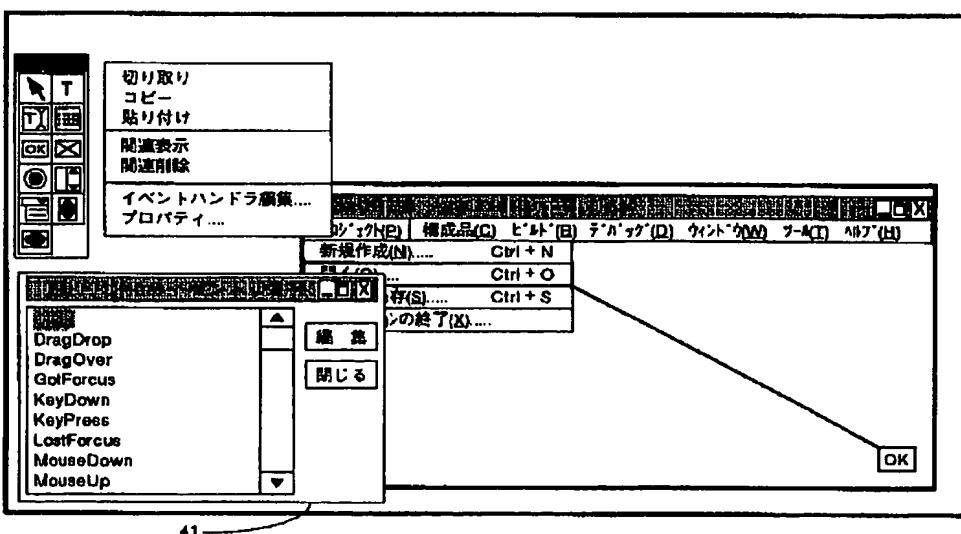


31

【図9】

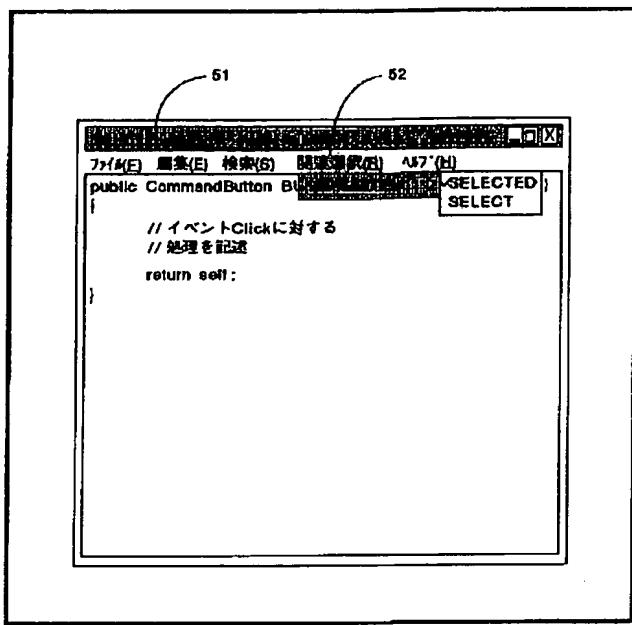


【図4】

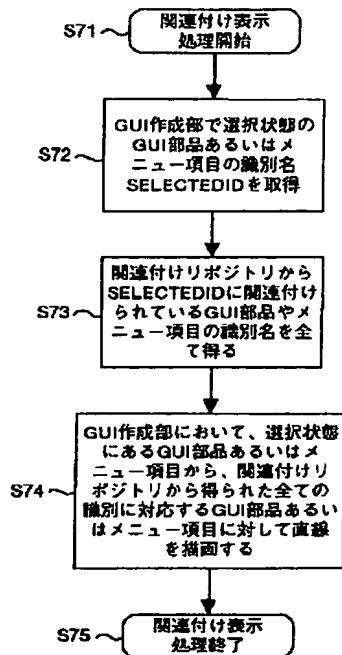


41

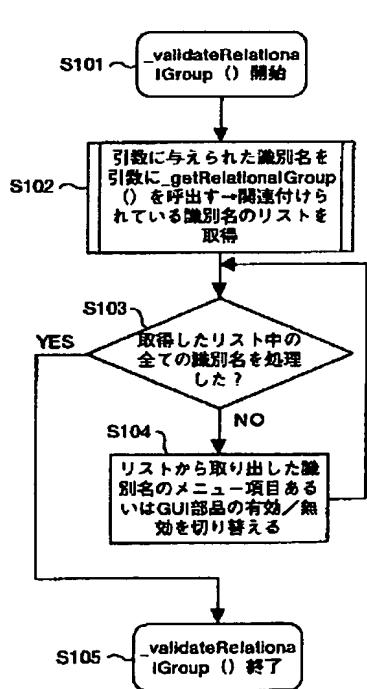
【図5】



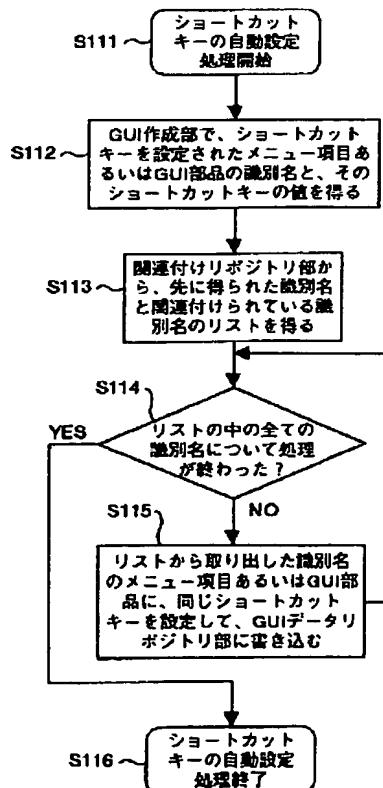
【図7】



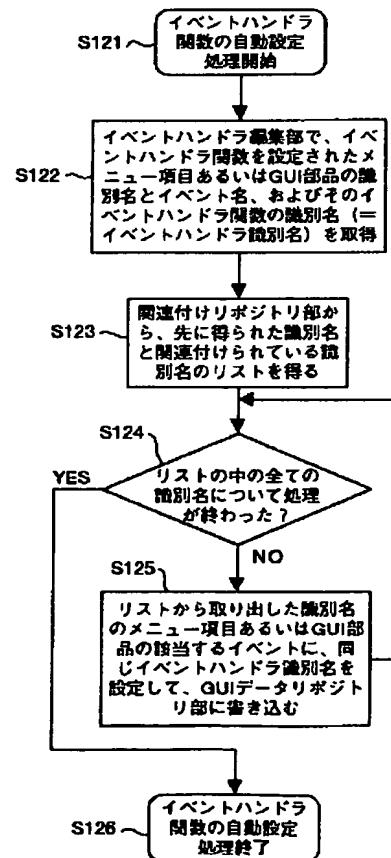
【図10】



【図11】



【図12】



フロントページの続き

(51) Int. Cl.⁶

G 09 G 5/14

識別記号

F I

G 09 G 5/14

Z